

SEPM

To understand DevOps: Principles, Practices and Devops engineer role and responsibilities

What is DevOps?

DevOps is a collaborative approach where teams work together to build & deliver secure software efficiently. It combines software development (dev) and operations (ops) to accelerate delivery through automation, collaboration, fast feedback and iterative improvement. Built on Agile methodology, DevOps creates a culture of accountability, collaboration & shared responsibility for business outcomes.

Core Principles of DevOps:

- Develop & test in production like environments
- Develop builds frequently
- Continuously develop operational quality

Key Practices of DevOps:

- Develop & test in production like environment
- Develop 1. Continuous Deployment

Continuous delivery and deployment originate from continuous integration, a method to rapidly develop, build & test new code with automation so that only code that is known to be good becomes part of a software product.

2. Continuous Development

This is the phase that involves planning & coding, versioning & managing builds of the software application functionality. Eg: Git, Github, Maven.

3: Continuous Testing:

Continuous Testing is executing automated tests, continuously & repeated against the code base & the various deployment environments. It is a software testing methodology which focuses on achieving continuous quality & improvement

Eg: Appium, Bamboo

4 Continuous Integration:

Continuous Integration refers to the build & unit testing stages of the software release process. Every revision that is committed triggers an automated build & test

Eg: Jenkins, Travis CI

5 Infrastructure Management

Without automation, building & maintaining large-scale modern without automation IT systems can be a resource intensive undertaking and can lead to increased risk due to manual error. Configuration & resource management is an automated method for maintaining computer systems & software in a known, consistent state.

6 Configuration Management:

Infrastructure as code is the practice of describing all software runtime environment & networking settings & parameters in simple textual format, that can be stored in your version control systems (VCS) & versioned on request. These test files are called

manifests & are used by DevOps tools to automatically provision & configure build servers, testing, staging & production environments.
Eg: chef, saltstack.

DevOps Engineer Role:

A DevOps engineer manages a company i.e. its infrastructure bridging development & operation, the primary goal is to improve the process and efficiency throughout the software development lifecycle.

Key Role:

1. **Facilitator of collaboration**
Bridging the gap between development, operations & QA teams to streamline communication.
2. **Automation specialist:**
Automate repetitive tasks like testing, deployment & monitoring.
3. **Continuous Integration & Continuous Delivery (CI/CD):**
Design, implement & maintain CI/CD pipelines to enable faster, reliable & repeatable software releases.
4. **Infrastructure as code**
Use tools like terraform, ansible or cloud formation to define & provision infrastructure through code.
5. **Monitoring & Incident Management**
Set up monitoring system to track application performance and troubleshoot issues in real time. It also ensures that systems are resilient and downtime is minimized.

6. Cloud & Infrastructure Management
Deploy, manage & optimize applications on cloud platform like AWS, Azure or Google Cloud, also handles container orchestration

Key Responsibilities:

1. Collaboration & Planning
Work with development & operations teams to plan & design scalable solutions
2. Configuration Management
Uses tools like puppet, chef or Ansible to manage server configuration & ensure consistency
3. Pipeline Management
Maintain CI/CD pipelines to ensure seamless build, test & deployment workflows
4. Monitoring & logging
Implement monitoring tools like Prometheus, Grafana or Splunk to track system health & measurement performance.
5. Support and Troubleshooting
Respond to incidents & resolve production issues promptly & identify root causes of failure & implement fixes
6. Documentation & Reporting
Document system configurations, deployment processes & troubleshooting guidelines.